



NoSQL

PROGRAMAÇÃO WEB 1

Objetivos de aprendizagem

- Introduzir o conceito de bancos de dados NoSQL
- Iniciar a utilização do MongoDB

Agenda

- SQL x NoSQL
- MongoDB
- Aplicações
- Benefícios
- Implantação / Configuração

SQL

- RDBMS (*Relational Database Management System*)
- Dados estruturados em tabelas relacionadas entre si
- Acesso e manipulação dos dados é feita utilizando SQL (*Structured Query Language*)

SQL

Vantagens

- Abstração de dados independente de implementação
- Dados otimizados para armazenamento em disco
- Utilização de recursos otimizada
- Maior consistência e integridade dos dados (atomicidade, transações, durabilidade, etc)

Desvantagens

- Necessidade de planejamento prévio da estrutura de dados, dificuldade em modificações
- Modelos de dados rígidos

Quando usar SQL?

- Recomendado para dados com relacionamento **bem definido**
- Quando a **integridade** de dados é crítica

NoSQL

- *Not only SQL*
- Desenvolvido na década de 1990, mas evidenciado em 2000 com a evolução da Computação em Nuvem, Internet, uso de dispositivos móveis e similares
- Surgiu da necessidade de bancos de dados com **menores restrições e mais escaláveis**

NoSQL

Vantagens

- Alta escalabilidade eliminando pontos de falha únicos
- Modelos de dados flexíveis
- Criação e mudança podem ser feitas *“on the fly”*
- Alta performance devido a simplicidade da implementação comparada a RDBMS

Desvantagens

- Menor disponibilidade de funcionalidades relacionadas a confiabilidade dos dados

Quando usar NoSQL?

- Conjuntos de dados mais flexíveis que não se enquadram ou não necessitam de relacionamentos
- Dados não estruturados
- Acesso rápido a grandes volumes de dados

Exemplos

SQL

- MySQL
- Oracle
- PostgreSQL
- DB2
- SQL Server

NoSQL

- Redis
- Fauna
- CouchDB
- Cassandra
- Elasticsearch
- **MongoDB**

“*MongoDB is an open source, nonrelational database management system (DBMS) that uses flexible documents instead of tables and rows to process and store various forms of data*”

MongoDB

<https://www.mongodb.com/pt-br/what-is-mongodb>

Características

- Unidade básica de dado consiste em documentos ou coleções de documentos
- Formatação básica em **JSON**
- Flexibilidade total na criação/consulta dos registros
- Confiável
- Altamente escalável

Base de documentos

- Um registro é chamado de documento
- Os documentos são estruturados em pares campo/valor, similar a **JSON**

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

“

A JSON *object* is an unordered set of **name/value pairs**.
An object begins with {*left brace* and ends with }*right brace*.
Each name is followed by :*colon* and the name/value pairs
are separated by ,*comma*.

”

<https://www.json.org/json-en.html>

Objeto JSON

Estrutura de documentos

- Os nomes de campos são *strings*
- Os valores podem assumir diversos tipos, inclusive outros documentos e *arrays* de documentos
- Tipos BSON (*Binary encoded JSON*)

```
var mydoc = {  
  _id: ObjectId("5099803df3f4948bd2f98391"),  
  name: { first: "Alan", last: "Turing" },  
  birth: new Date('Jun 23, 1912'),  
  death: new Date('Jun 07, 1954'),  
  contribs: [ "Turing machine", "Turing test", "Turingery" ],  
  views : NumberLong(1250000)  
}
```

BSON Types

Formato Binário de Serialização

<https://www.mongodb.com/docs/manual/reference/bson-types/>


```
{
  _id: <ObjectId>,
  username: "123xyz",
  contact: {
    phone: "123-456-7890",
    email: "xyz@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}
```

Embedded sub-document

Embedded sub-document

ObjectId

- MGD requer que cada documento de uma coleção possua um campo **_id** único que funciona como chave primária
- Se um documento omite o **_id**, o valor é gerado automaticamente
- **_id** é um nome de campo reservado

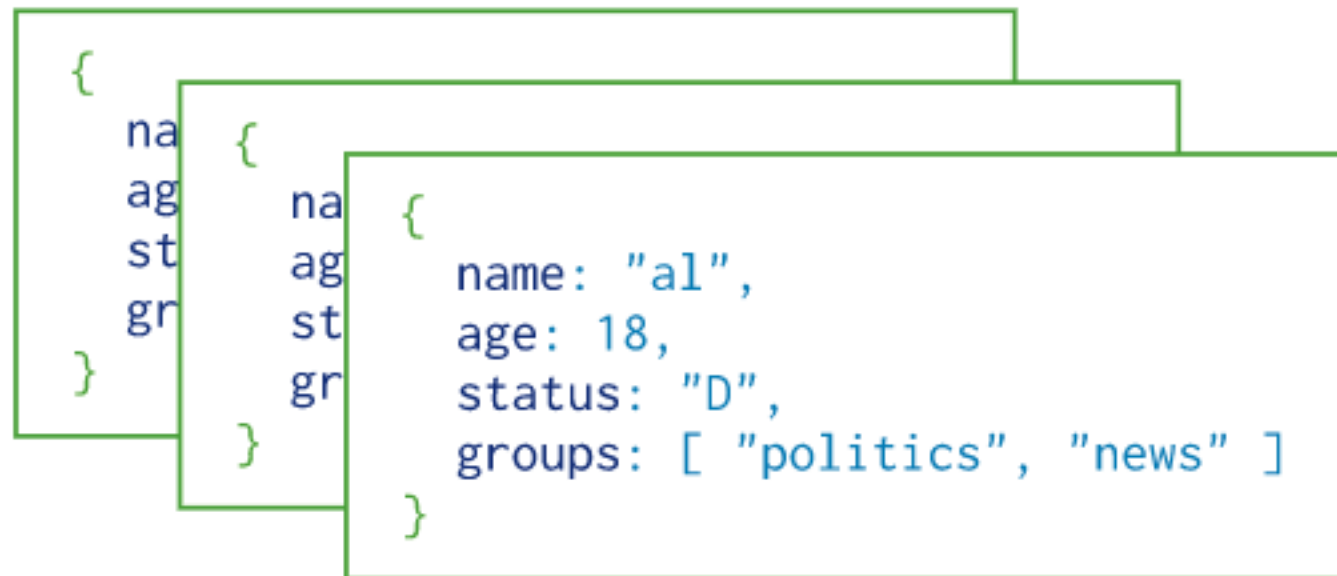
```
var mydoc = {  
    _id: ObjectId("5099803df3f4948bd2f98391"),  
    name: { first: "Alan", last: "Turing" },  
    birth: new Date('Jun 23, 1912'),  
    death: new Date('Jun 07, 1954'),  
    contribs: [ "Turing machine", "Turing test", "Turingery" ],  
    views : NumberLong(1250000)  
}
```

Por que utilizar documentos?

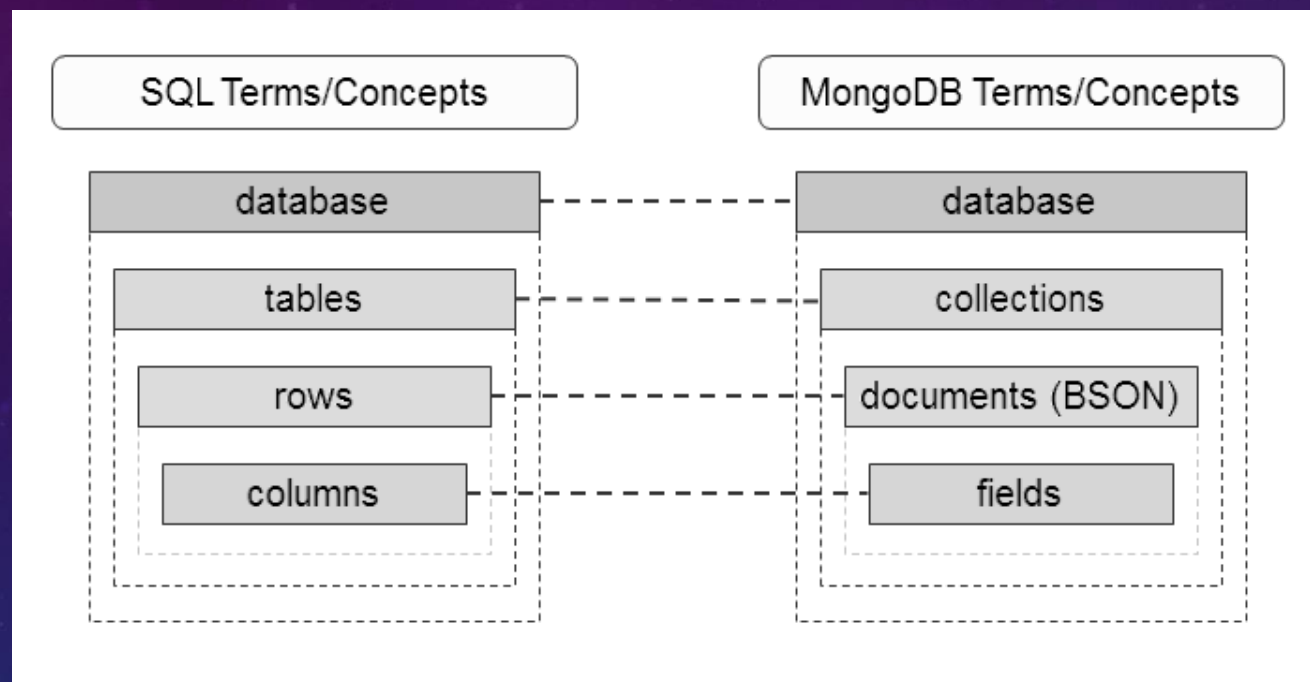
- Praticamente, correspondem aos **dados nativos** utilizados nas linguagens, logo são facilmente processáveis
- Polimorfismo de dados é facilitado
- Reduzem a necessidade de **joins** de alto custo computacional

Coleções

- Um conjunto de documentos é chamado de coleção (*collection*)
- *Collections* são equivalente a tabelas em RDBMS



Collection



CRUD

ACESSANDO O MONGODB

CRUD

- *Create*
- *Read*
- *Update*
- *Delete*

Create

- `db.collection.insertOne()`
- `db.collection.insertMany()`

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,            ← field: value
  status: "pending"   ← field: value } document
})
```


Read

- `db.collection.find()`

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier

Read

MongoDB Shell

```
db.inventory.find({})
```

```
db.inventory.find(  
  { status: "D" } )
```

```
db.inventory.find(  
  { $or: [ { status: "A" }, { qty:  
    { $lt: 30 } } ] } )
```

SQL

```
SELECT * FROM inventory
```

```
SELECT * FROM inventory WHERE status =  
"D"
```

```
SELECT * FROM inventory WHERE status =  
"A" OR qty < 30
```

Update

- `db.collection.updateOne(<filter>, <update>, <options>)`
- `db.collection.updateMany(<filter>, <update>, <options>)`
- `db.collection.replaceOne(<filter>, <update>, <options>)`

Delete

- `db.collection.deleteMany()`
- `db.collection.deleteOne()`

Exercício – Parte 1

- Seguir o tutorial abaixo para realizar operações diversas utilizando o Mongo Shell (mongosh):
 - <https://www.mongodb.com/docs/v7.0/tutorial/getting-started/>

“ MongoDB Atlas is a multi-cloud database service by the same people that build MongoDB. Atlas simplifies deploying and managing your databases while offering the versatility you need to build resilient and performant global applications on the cloud providers of your choice.”

<https://www.mongodb.com/docs/atlas/>

MongoDB Atlas

- Criar conta Atlas
 - <https://www.mongodb.com/docs/atlas/tutorial/deploy-free-tier-cluster/>
- Implantar um cluster gratuito (clique em **Atlas UI** no link abaixo)
 - <https://www.mongodb.com/docs/atlas/tutorial/deploy-free-tier-cluster/>
- Configurar o IP para acesso
 - <https://www.mongodb.com/docs/atlas/security/add-ip-address-to-list/>

Exercício – Parte 2

- Criar um novo banco de dados (`sample_data`) no seu cluster recém criado
 - Deployment->Database->Browse Collections->Add My Own Data
- Em **`sample_data`** criar a coleção **`students`**
- Adicionar documentos na coleção segundo o modelo a seguir

Exercício – Parte 2

```
{  
  "name": "Jason Ray",  
  "profession": "Software Engineer",  
  "age": 31,  
  "address": {  
    "city": "New York",  
    "postalCode": 64780,  
    "Country": "USA"  
  },  
  "languages": ["Java", "Node.js", "JavaScript", "JSON"],  
  "socialProfiles": [  
    {  
      "name": "Twitter",  
      "link": "https://twitter.com"  
    },  
    {  
      "name": "Facebook",  
      "link": "https://www.facebook.com"  
    }  
  ]  
}
```

Exercício – Parte 3

1. Criar o banco de dados *db*
2. Adicionar o conteúdo do arquivo **restaurants.json** (o arquivo disponibilizado na **Aula 17**) a coleção com nome *restaurants* a ser criada

Exercício – Parte 3

3. Para importar (ou exportar) **dados** (*Data Import and Export Tools*) de um arquivo externo siga os passos:

- Database->Browse collections->Cmd Line Tools
- É necessário baixar o MongoDB Database Tools
- *Data Import and Export Tools* (siga as instruções)

Exercício – Parte 4

- Depois de criado o banco de dados db e adicionado a coleção **restaurants** o arquivo **restaurants.json** (disponibilizado na Aula 17), responda às questões propostas no exercício abaixo
 - <https://www.w3resource.com/mongodb-exercises/>

Referências

- Links úteis para realizar as ações abaixo via Node.js
 - INSERT
 - <https://www.mongodb.com/docs/manual/tutorial/insert-documents/>
 - QUERY
 - <https://www.mongodb.com/docs/manual/tutorial/query-documents/>
 - UPDATE
 - <https://www.mongodb.com/docs/manual/tutorial/update-documents/>
 - REMOVE
 - <https://www.mongodb.com/docs/manual/tutorial/remove-documents/>

Referências

- <https://www.ibm.com/cloud/learn/mongodb>
- <https://www.mongodb.com/pt-br/what-is-mongodb>
- <https://www.mongodb.com/basics/bson>
- <https://www.json.org/json-en.html>